



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**NÁVRH CHYTRÉHO SENZORU S KOMUNIKAČNÍM  
ROZHRANÍM ZALOŽENÉM NA TECHNOLOGII LORAWAN**

DESIGN OF SMART SENSOR EQUIPPED WITH LORA TECHNOLOGY

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Štěpán Šigut**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Martin Štůsek**

**BRNO 2018**

# Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

**Student:** Štěpán Šigut

**ID:** 173755

**Ročník:** 3

**Akademický rok:** 2017/18

## NÁZEV TÉMATU:

### Návrh chytrého senzoru s komunikačním rozhraním založeném na technologii LoRaWAN

## POKyny PRO VYPRACOVÁNÍ:

Tato bakalářská práce se bude zabývat návrhem inteligentního senzoru založeného na technologii LoRa (Long Range) umožňující měření základních fyzikálních veličin (teplota, tlak, vlhkost). Praktickým výstupem práce bude návrh a výroba desky plošných spojů včetně praktického ověření funkčnosti. Důraz bude kladen zejména na možnost bateriového napájení s dlouhou životností v řádu měsíců až jednotek let, dle četnosti zpráv. Kromě funkčního senzoru bude výstupem bakalářské práce také webové uživatelské rozhraní. To bude umožňovat vizualizaci aktuálních ale i historických hodnot v odpovídajícím formátu. Funkčnost vytvořené platformy bude ověřena v reálné LoRaWAN síti na VUT v Brně.

## DOPORUČENÁ LITERATURA:

[1] ROWLAND, Claire, Elizabeth GOODMAN, Martin CHARLIER, Ann LIGHT a Alfred LUI. Designing connected products: UX for the consumer Internet of things. Sebastopol: O'Reilly, 2015. ISBN 978-1449372569.

[2] CHOU, Timothy. Precision: Principles, Practices and Solutions for the Internet of Things. Crowdstory, 2016. ISBN 9781329843561.

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 29.5.2018

**Vedoucí práce:** Ing. Martin Štůsek

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## ABSTRAKT

Rostoucí kapacita baterií, vyšší integrace polovodičů a klesání ceny za výpočetní výkon podporují populární myšlenku internetu věcí. Poslední roky lze pozorovat rychle rostoucí trend v počtu připojených zařízení do internetu. To klade velké nároky na technologie bezdrátových sítí, které hrají každým rokem významější roli na technologickém trhu. V posledních letech zároveň vzniklo mnoho nových bezdrátových technologií, které se snaží řešit některá úskalí masového nasazení internetu věcí. Mezi tyto nové technologie patří stále ještě relativně mladá technologie LoRaWAN, které se tato práce věnuje.

## KLÍČOVÁ SLOVA

IoT, LoRa, LoRaWAN, LPWAN, bezdrátové sítě, chytrá města, chytrý dům

## ABSTRACT

Over the past years few trends have become as profound in the technology industry as yearly increases in battery capacity, transistor density and price drops for computing power. These trends follow a steep curve and drive what has become known as IoT or Internet of Things. Ever increasing number of devices connected to the internet are putting high demands on wireless network technologies. During the last decade several new wireless technologies have emerged and are competing to become the next go-to technologies for applications such as smart cities, smart cars and smart factories. LoRaWAN is one relatively new competitor to the market and is the central topic of this thesis.

## KEYWORDS

IoT, LoRa, LoRaWAN, LPWAN, wireless networks, smart cities, smart home

ŠIGUT, Štěpán. *Návrh chytrého senzoru s komunikačním rozhraním založeném na technologii LoRa*. Brno, 2017, 48 s. Semestrální projekt. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Martin Štůsek

## PROHLÁŠENÍ

Prohlašuji, že svůj semestrální projekt na téma „Návrh chytrého senzoru s komunikačním rozhraním založeném na technologii LoRa“ jsem vypracoval(a) samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto semestrálního projektu jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce panu Ing. Martinovi Štůskovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)

# OBSAH

<b>Úvod</b>	<b>11</b>
<b>1 Bezdrátové sítě v internetu věcí</b>	<b>12</b>
1.1 Wi-Fi IEEE 802.11 . . . . .	12
1.2 IEEE 802.15.4 . . . . .	13
1.3 Sigfox . . . . .	13
1.4 NB-IoT . . . . .	14
<b>2 LoRaWAN technologie</b>	<b>15</b>
2.1 LoRa . . . . .	15
2.1.1 Modulace s rozprostíráním spektra . . . . .	16
2.2 Způsob komunikace . . . . .	19
2.3 Možnosti využití . . . . .	20
2.3.1 Odečet dat v reálném čase . . . . .	20
2.3.2 Chytrá města . . . . .	20
<b>3 Realizace LoRaWAN zařízení</b>	<b>22</b>
3.1 Meteorologická jednotka . . . . .	22
3.1.1 MCU Arduino . . . . .	22
3.1.2 LoRaWAN modul SX1276 . . . . .	25
3.1.3 Senzory meteorologických veličin . . . . .	25
3.1.4 Rozhraní SPI a I <sup>2</sup> C . . . . .	25
3.1.5 Implementace koncového zařízení . . . . .	26
3.1.6 Režim snížené spotřeby . . . . .	30
3.2 Síťový server LORIoT . . . . .	31
3.3 Aplikační server . . . . .	32
3.3.1 WebSocketový klient . . . . .	32
3.3.2 Databáze . . . . .	32
3.3.3 Datový server . . . . .	33
3.4 Interpretace zasílaných dat . . . . .	35
<b>4 Závěr</b>	<b>37</b>
<b>Literatura</b>	<b>38</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>41</b>
<b>Seznam příloh</b>	<b>43</b>

A	Meteorologická jednotka	44
B	Aplikační server	45
C	Interpretace a zobrazení dat	47

# SEZNAM OBRÁZKŮ

2.1	DSSS modulace signálu pomocí pseudonáhodného kódu. . . . .	16
2.2	DSSS demodulace signálu. . . . .	17
2.3	Čerp pulz v časové oblasti. . . . .	18
2.4	Čerp pulz ve frekvenční oblasti. . . . .	18
2.5	Časový průběh změny frekvence CSS signálu. . . . .	18
3.1	Architektura systému pro měření meteorologických veličin. . . . .	23
3.2	Blokové schéma meteorologické jednotky. . . . .	26
3.3	Elektronické schéma zapojení součástek jednotky. . . . .	27
3.4	Návrh desky plošných spojů. . . . .	28
3.5	Vyhotovená deska plošných spojů. . . . .	29
3.6	Stránka s meteorologickými daty ve formě grafů. . . . .	36



# SEZNAM TABULEK

2.1	Maximální propustnost LoRaWAN sítě. . . . .	19
3.1	Velikost paměti v MCU ATmega328. . . . .	24
3.2	Porovnání JSON a vlastního formátu předávaných dat. . . . .	30

# SEZNAM VÝPISŮ

A.1	Nastavení v kódu meteorologické jednotky. . . . .	44
B.1	Směrovač HTTP dotazů knihovny Flask. . . . .	45
B.2	Databázové modely knihovny SQLAlchemy. . . . .	46
C.1	HTML s JavaScript kódem pro zobrazení zpráv z meteorologické jednotky. . . . .	47

# ÚVOD

V současnosti a blízké historii lze v elektrotechnickém průmyslu sledovat několik trendů. Jedním z nich je minimalizace velikosti a spotřeby elektronických zařízení. Minimalizace a snaha zjednodušit práci elektroinženýrů vedla k značnému rozšíření SoC (systém na čipu - System on Chip). Shodou okolností se už mnoho let daří plynulým tempem zvyšovat maximální kapacitu baterií. Tyto faktory společně se snižováním ceny za výpočetní výkon vytvořily optimální prostředí pro vznik levných senzorických zařízení připojených do bezdrátové sítě. Myšlenka využívat tyto senzorické zařízení pro zvýšení efektivity průmyslu, monitorování špatně dostupných oblastí, ale i integrované do předmětů denní potřeby si našla své příznivce, a tak vzniklo to, co lze označit pojmem internet věcí.

IoT (internet věcí - Internet of Things) je jedno z nejdiskutovanějších témat současného průmyslu. Představa chytrých měst, domácností a automobilů (*Smart Cities*, *Smart Homes*, *Smart Cars*) živí diskuze jak v profesionální inženýrské scéně, tak ve veřejných médiích. Mediální nadšení je podpořeno i několika odbornými autory, kteří do roku 2025 očekávají zásadní nárůst v počtu zařízení připojených k internetu [1][2].

Velký nárůst v počtu zařízení s potřebou bezdrátové komunikace s sebou přináší řadu problémů pro návrh bezdrátových sítí, do kterých jsou zařízení připojena. Mezi specifické nároky na bezdrátové sítě pro IoT patří nízká spotřeba elektrické energie při vysílání a příjmu a potlačení interferencí při velkém množství komunikujících zařízení. Právě z důvodu nutnosti nízké spotřeby během vysílání tyto sítě nazýváme LPWAN (Low-Power Wide-Area Network).

Tato práce se věnuje popisu LPWAN sítí, včetně praktické implementace bezdrátového senzoru založeného na technologii LoRaWAN (Long Range Radio Wide Area Network) využívající modulaci LoRa (Long Range Radio).

# 1 BEZDRÁTOVÉ SÍTĚ V INTERNETU VĚCÍ

Jak bylo zmíněno v úvodu, požadavky na velké množství připojených zařízení do jedné bezdrátové sítě a jejich nároky na energetickou úspornost představují překážku pro aktuální bezdrátové technologie jako Wi-Fi, mobilní sítě 2G/3G/LTE (Long Term Evolution) nebo Bluetooth, které jsou dnes považovány za průmyslový standard. Nedostatky těchto sítí se snaží řešit technologie, které bychom zařadili do kategorie LPWAN. V následujících sekcích budou popsány jak tradiční bezdrátové technologie jako Wi-Fi, tak existující LPWAN technologie z pohledu jejich využitelnosti pro IoT a nedostatků, které omezují jejich široké nasazení. Jako kritické vlastnosti pro nasazení technologie v IoT byly vybrány tyto faktory:

- Nízká spotřeba energie během komunikace.
- Dostatečně velký dosah radiové komunikace.
- Dostatečná rychlost přenosu.
- Možnost připojit velké množství zařízení do jedné sítě.

## 1.1 Wi-Fi IEEE 802.11

Standard Wi-Fi je v dnešní době nejrozšířenější technologií pro bezdrátovou komunikaci. Standardy 802.11b a 802.11g operují v nelicencovaném pásmu 2,4 GHz. Ačkoliv je Wi-Fi v oblasti IoT využívána například pro domácí automatizaci není pro nasazení mimo uzavřené prostory vhodnou technologií. Standard byl navržen pro připojení osobních počítačů s prioritou vysoké přenosové rychlosti a z tohoto důvodu se vyznačuje nízkým dosahem a je velmi energeticky náročný. Energetickou náročnost lze demonstrovat na SoC ESP8266, který pochází z laboratoří společnosti Espressif a je cílený právě pro použití v nízkoenergetických vestavěných systémech. Dle výrobce se průměrná spotřeba během vysílání pohybuje v rozmezí 120–170 mA dle vybraného 802.11 standardu a vysílacího výkonu a spotřeba během příjmu je okolo 50 mA [3]. Vysoká spotřeba omezuje využití bateriově napájených Wi-Fi zařízení prakticky pouze pro spotřební elektroniku.

Pro využití v IoT je však perspektivní standard 802.11ax, který sice neřeší problematiku vysoké spotřeby koncových zařízení, nicméně kromě zvýšení přenosových rychlostí je jeho hlavním přínosem využití techniky přístupu k médiu OFDMA (Orthogonal Frequency Division Multiple Access). Tato technologie umožňuje k jednomu přístupovému bodu připojit výrazně větší počet koncových zařízení a zajišťuje dynamické přidělení přenosového pásma koncovým zařízeníům [4]. Masovější rozšíření technologie 802.11ax je plánováno na rok 2019 [5].

## 1.2 IEEE 802.15.4

Institut IEEE (Institut pro elektrotechnické a elektronické inženýrství - Institute of Electrical and Electronics Engineers) vydal standard 802.15.4 popisující LPWAN síť v roce 2003 [6]. Standard nepředstavuje přímo využitelnou technologii, protože nedefinuje všechny vrstvy referenčního modelu ISO/OSI (mezinárodní organizace pro normalizaci / propojení otevřených systémů - International Organization for Standardization / Open Systems Interconnection), ale definuje pouze fyzickou a spojovou vrstvu referenčního modelu. Standard také nepopisuje konkrétní způsob komunikace, velikost zpráv, šifrování a konkrétní implementaci bezdrátové technologie už nechává na jednotlivých ze standardu vycházejících technologiích. Typickým zástupcem této kategorie a konkrétní implementace standardu 802.15.4 je technologie ZigBee, která operuje převážně v nelicencovaném pásmu 2,4 GHz, které je sdílené s technologií Wi-Fi, Bluetooth a dalšími méně rozšířenými bezdrátovými standardy. Některá zařízení založená na technologii ZigBee však operují i v pásmech 868 MHz (v Evropě), 780 MHz (v Asii) a 915 MHz (v USA a v Austrálii) [7]. Hlavním problémem, který brání vyššímu rozšíření sítí založených na technologii ZigBee je velká míra interferencí způsobená silně zarušeným nelicencovaným pásmem a malým počtem vysílacích kanálů definovaných v standardu [8].

Standard 802.15.4 je podporován většinou velkých výrobců integrovaných obvodů, jako jsou společnosti Texas Instruments, Microchip, Atmel, NXP a další [9][10][11]. Společným znakem SoC podporujících standard 802.15.4, které tyto společnosti vyrábějí je podpora více než jedné technologie využívající 802.15.4 standard - typicky se jedná o technologie ZigBee, Thread, WirelessHART a další.

## 1.3 Sigfox

Sigfox je proprietární technologie stejnojmenné francouzské společnosti. Snahou technologie je celoplošné pokrytí pomocí buněk (základnových stanic) podobně, jako tomu je u architektury stávajících mobilních sítí. Technologie operuje s celkovou šířkou pásma 200 kHz v nelicencovaném pásmu 868 MHz (v Evropě) a 915 MHz (v USA) a pro přenos jedné zprávy využívá velmi úzké pásmo 100 Hz. Pásmo je tedy rozděleno celkový počet 2000 kanálů. To umožňuje na celém využívaném spektru 200 kHz vysílat z mnoha zařízení zároveň, ale zároveň je tím omezena maximální přenosová rychlost sítě na pouze 100 bit/s, což pro i řadu aplikací zdánlivě nenáročných na přenosovou rychlost může představovat omezení.

Nevýhodou technologie je nezasílání potvrzovacích rámců základnovou stanicí a absence systému na detekci kolizí ve vysílání. Technologie se případné kolize ve vysílání snaží řešit tím, že každá zpráva je odeslána třikrát, vždy na jiném kanálu

s jinou frekvencí. S rostoucím množstvím zařízení a interferencemi z jiných zařízení operujících ve stejném nelicencovaném pásmu se však dá předpokládat snížení propustnosti celé sítě a ztrátovost paketů, kterou pro chybějící mechanismus potvrzení zpráv, lze velmi těžko detekovat [8],[12]. Pravděpodobnost výskytu kolize se s rostoucím počtem zařízení výrazně zvyšuje a omezuje to možnost koexistence Sigfox technologie vedle ostatních LPWAN sítí [16].

## 1.4 NB-IoT

Technologie NB-IoT (NarrowBand IoT) patří mezi technologie, které jsou zatím méně rozšířené, nicméně velmi rychle získávají na popularitě. Obrovská výhoda této technologie však spočívá v tom, že operuje na vyhrazeném frekvenčním pásmu, které není na rozdíl od sdílených pásem zarušené a zařízení podporující tuto technologii tedy budou profitovat z lepších rádiových podmínek komunikačního kanálu. Standard je definován pracovní skupinou 3GPP (3rd Generation Partnership Project), která je zároveň autorem standardu LTE, ze kterého technologie NB-IoT vychází a vedle technologie NB-IoT skupina definuje ještě další standard pro IoT využití - LTE-M.

Licencovaná pásma jsou v převážné většině využívána mobilními operátory, kteří v licencovaném pásmu provozují svoje stávající mobilní komunikační sítě. Technologie NB-IoT byla proto navržena tak, aby mohla snadno koexistovat se současnými mobilními sítěmi a neomezovala současný provoz. Kvůli těmto požadavkům umí technologie operovat jak ve stávajícím LTE pásmu, tak také v ochranných pásmech LTE kanálů, což nesnižuje propustnost současných sítí [13]. Technologie využívá šířku pásma 180 kHz a maximální přenosová rychlost je 250 kbit/s což přesahuje možnosti jak LoRaWAN, tak Sigfox technologií. Většímu rozšíření zatím bránila vyšší cena komunikačních modulů a pomalejší adopce technologií mobilními operátory [14], nicméně v současnosti je ekosystém dodavatelů a provozovatelů již rozšířenější, na území České Republiky pak převážně díky společnosti Vodafone, která jako první uvedla do provozu NB-IoT síť [15].

## 2 LORAWAN TECHNOLOGIE

LoRaWAN je jednou z nejrozšířenějších technologií v oblasti LPWAN. Technologie byla vyvinuta společností Cycleo - francouzskou společností, která se později v rámci akvizice stala součástí společnosti Semtech, která je zároveň současným výrobcem LoRaWAN kompatibilních modulů dostupných jako integrované obvody nebo SoM (systém na modulu - System on Module), které jsou vhodné k přímému využití v koncových zařízeních. LoRaWAN podobně jako konkurenční LPWAN sítě využívá nelicencované pásmo 868 MHz (v Evropě), 915 MHz (v USA) a 433 MHz (v Asii). Každá LoRaWAN síť obsahuje dva typy zařízení:

- Koncová zařízení.
- Hraniční brány.

Topologie sítě je hvězda, kde koncová zařízení komunikují přímo s hraniční bránou (komunikace probíhá přes jediný skok) a hraniční brány jsou nejčastěji opět spojeny do hvězdy se síťovým serverem, který sbírá data z koncových zařízení. Spojení výchozí brány se síťovým serverem probíhá přes síť založenou na jiné komunikační technologii, nejčastěji pak přes Ethernet nebo mobilní síť. Koncová zařízení jsou nejčastěji bateriově napájena a je u nich tudíž kladen velký důraz na výdrž baterie a také na nízkou cenu. Ačkoliv technologie umožňuje obousměrnou komunikaci, upřednostňována je komunikace z koncových zařízení směrem k síťovému serveru právě z důvodu nižších energetických nároků zařízení, které nemusí být neustále připravené k přijmutí zpráv, ale mohou využívat režim spánku, který šetří elektrickou energii, když zařízení zrovna nevysílá.

Následující sekce popisují LoRaWAN technologii detailněji z pohledu použité modulace, způsobu komunikace a praktického využití LoRaWAN sítí.

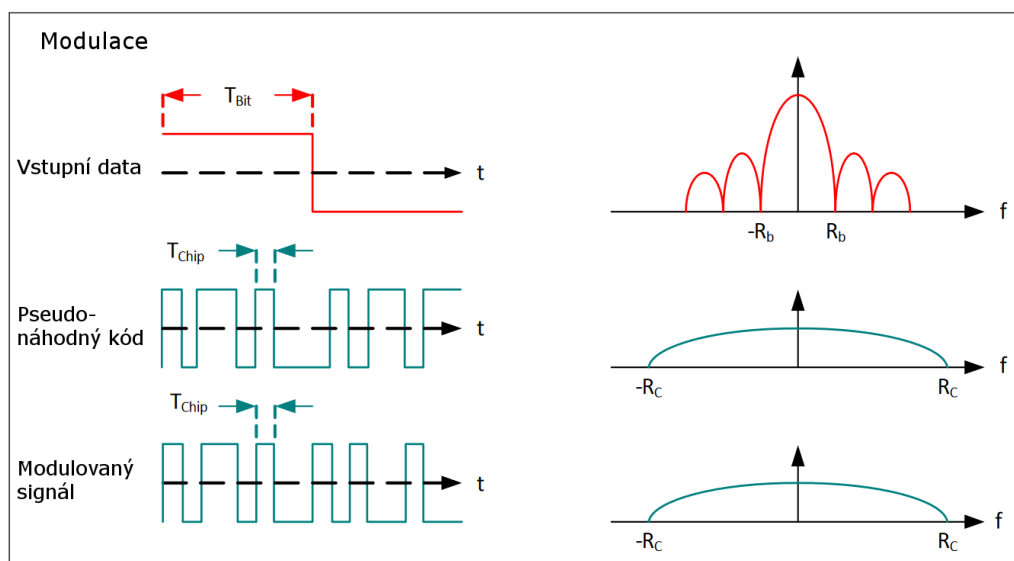
### 2.1 LoRa

Názvem LoRa je označen souhrn technologií použitých ve fyzické vrstvě LoRaWAN sítí. LoRa se vyznačuje nízkou spotřebou energie při vysílání (výdrž koncových zařízení až 10 let), nízkou přenosovou rychlostí (např. 27 kb/s při použití  $SF = 7$  (činitel rozprostření - Spreading Factor) a šířce kanálu 500 kHz) a dlouhým dosahem komunikace (2–5 km v městských oblastech a až 15 km v oblastech bez městské zástavby).

LoRa pro evropské pásmo 868 MHz definuje 3 kanály o šířce 125 kHz, které jsou povinně podporovány všemi koncovými zařízeními a 3 další kanály o stejné šířce, jejichž implementace je volitelná. Pro přenos dat standard definuje *LoRa spread spectrum* modulaci, která je odvozena z CSS (Chirp Spread Spectrum) modulace a používá některé prvky DSSS (Direct Sequence Spread Spectrum) modulace [19].

### 2.1.1 Modulace s rozprostíráním spektra

CSS podobně jako další *spread spectrum* modulace používá metodu rozprostření spektra. Rozprostření spektra se obvykle používá k zvýšení odolnosti proti rušení a zvýšení citlivosti přijmače signálu. DSSS (Direct Sequence Spread Spectrum) a FHSS (Frequency Hopping Spread Spectrum) jsou dalšími zástupci *spread spectrum* modulací. Obě technologie využívají pro rozprostření spektra pseudonáhodné kódy předem známé oběma komunikujícím stranám. FHSS nejprve rozdělí komunikační kanál na mnoho podkanálů na kterých může probíhat komunikace a po zahájení komunikace podle předem známého mechanismu (kódu) mění komunikační podkanál. Tím FHSS dosahuje dobré odolnosti proti úzkopásmovému rušení. DSSS nejprve připraví pseudonáhodnou sekvenci bitů, tzv. chipů (využívány jsou např. Goldovy a Barkerovy kódy), která má vyšší modulační rychlost (čipovou rychlost) než data přenášené zprávy. Data přenášené zprávy jsou pak vynásobena připravenou pseudonáhodnou sekvencí bitů. Tento proces můžeme pozorovat na Obr. 2.1, kde v levé části lze vidět modulační proces v časové oblasti a v pravé části v oblasti frekvenční. Použitím pseudonáhodného kódu se do přenášených dat zavádí redundance a zvyšuje

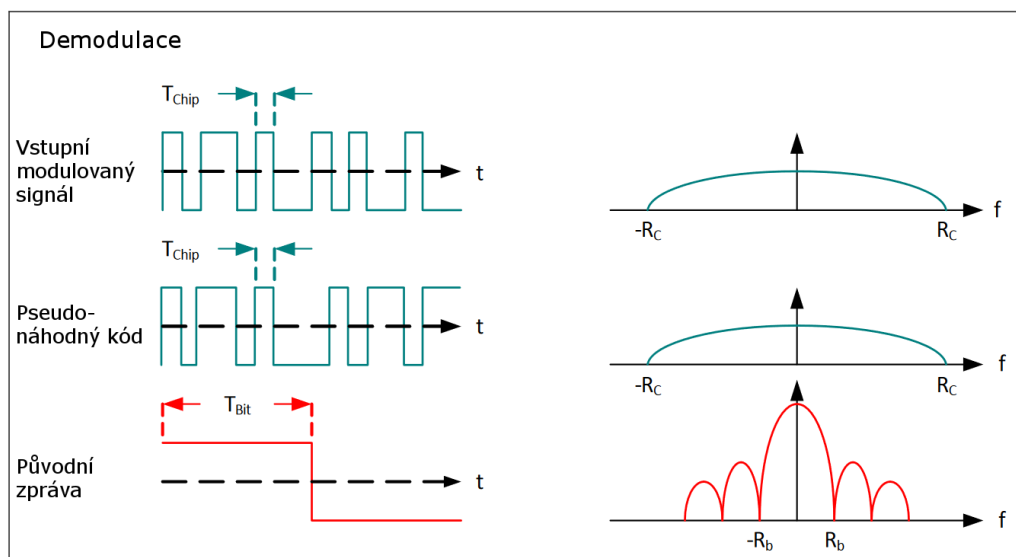


Obr. 2.1: DSSS modulace signálu pomocí pseudonáhodného kódu.

se šířka přenosového pásma, ale zároveň tato technika příznivě ovlivňuje schopnost přenášet data i v zarušeném prostředí a zvyšuje celkový dosah komunikace. Demodulace vysílaného signálu probíhá opačným postupem. Vysílaný signál přijímač vynásobí totožným pseudonáhodným kódem, který byl použit v procesu modulace a výsledkem jsou data původní zprávy. Demodulační proces lze vidět na obr. 2.2.

CSS se liší tím, že nevyužívá pseudonáhodného kódu, ale spoléhá se na lineární





Obr. 2.2: DSSS demodulace signálu.

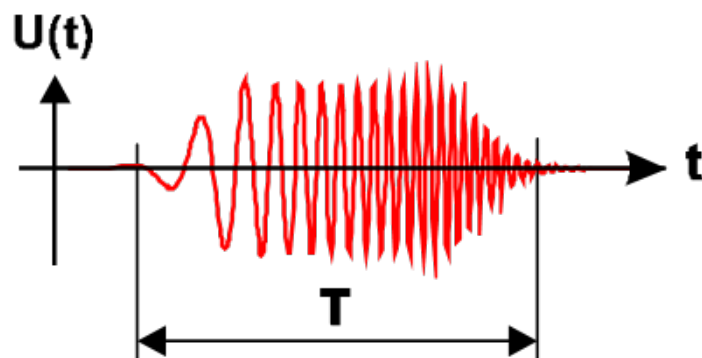
charakter krátkých pulzů tzv. čerpů (chirp). Čerp (chirp) je krátký sinusový pulz s lineárně se zvyšující či snižující frekvencí. Na obr. 2.3 je znázorněn čerp pulz v časové oblasti a na obr. 2.4 lze pozorovat čerp pulz ve frekvenční oblasti. Frekvence čerp pulzu se může měnit z nízké frekvence do vyšší, pak se jedná o tzv. Up-Chirp, nebo se frekvence mění v opačném směru a pak jde o tzv. Down-Chirp. Zajímavý grafický pohled na CSS modulaci lze pozorovat na obr. 2.5, který zobrazuje signál nesoucí zprávu modulovanou nejjednodušší CSS modulací v časově frekvenční oblasti. Použitá modulace rozlišuje jen dvě hladiny na symbol a každý symbol tedy nese maximálně jeden bit informace. Obr. 2.5 zobrazuje zprávu nesoucí bitovou sekvencí 1101, kde bit 1 je reprezentován up čerpem a bit 0 down čerpem. Technologie LoRa pro modulování zpráv využívá složitější modulaci, která je založena na velikosti změny frekvence v čerp pulzech [18].

Důležitou vlastností všech *spread spectrum* systémů je SF (činitel rozprostření - Spreading Factor). Činitel rozprostření udává poměr mezi čipovou rychlostí a modulační rychlostí a můžeme jej vypočítat podle vztahu 2.1, kde  $R_c$  je čipová rychlost a  $R_s$  je modulační rychlost.

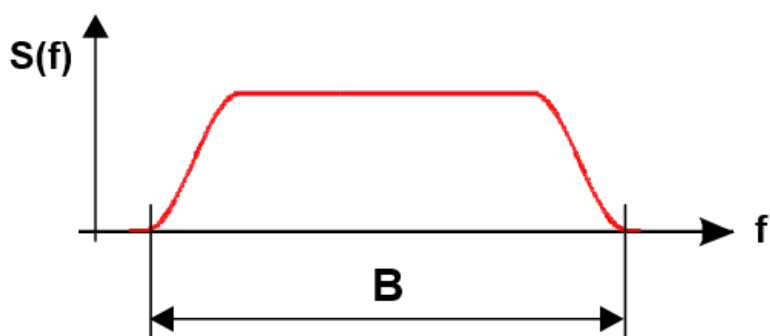
$$SF = R_c / R_s. \quad (2.1)$$

Vyšší činitel rozprostření znamená vyšší odolnost vůči rušení a maximální dosah signálu, ale z rovnice vyplývá, že při konstantní zvolené čipové rychlosti zvýšení činitele rozprostření znamená snížení modulační rychlosti a snižuje se tedy přenosová rychlost systému.

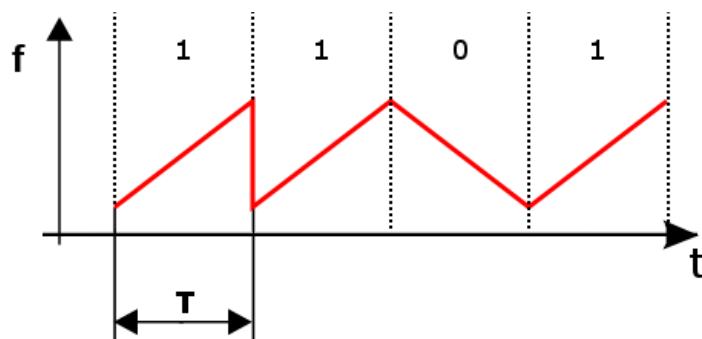
LoRa standard definuje šest různých SF (postupně od  $SF = 7$  až po  $SF = 12$ ). Kódy těchto SF jsou vzájemně ortogonální, což znamená, že v jednu chvíli může



Obr. 2.3: Čerp pulz v časové oblasti.



Obr. 2.4: Čerp pulz ve frekvenční oblasti.



Obr. 2.5: Časový průběh změny frekvence CSS signálu.

v síti komunikovat několik zařízení zároveň za předpokladu použití rozdílných SF pro každé zařízení. SF ovlivňuje také přenosovou rychlost LoRaWAN sítě, která se pohybuje od 22 bit/s do 27 kb/s.

## 2.2 Způsob komunikace

Dle standardu komunikace začíná *Join* procedurou, která probíhá na několika kanálech pomocí pseudonáhodného přeskočení mezi kanály. Další pakety jsou pak posílány pomocí CSS modulace popsané v sekci 2.1. Paket se skládá z preamble, hlavičky, datového obsahu (s maximální délkou 51–222 bytů podle použitého SF) a CRC (cyklický redundantní součet - Cyclic Redundancy Check) [17].

LoRaWAN definuje podle komunikačních schopností a spotřeby elektrické energie tři typy koncových zařízení - zařízení *třídy A*, *B* a *C*. Dle standardu musí všechna koncová zařízení podporovat alespoň *třídu A* a zařízení podporující vyšší třídu musí s *třídou A* zůstat kompatibilní.

Zařízení *třídy A* jsou energeticky nejméně náročné a pro přístup k síti používají mechanismus tzv. *prostá ALOHA*. To znamená, že koncové zařízení odešle rámec a pak čeká na odpověď ve dvou přijímacích oknech. Okna jsou vůči odeslanému rámci postupně časově posunutá nejčastěji o 1 a 2 sekundy. Komunikace z hraničních bran směrem ke koncovým zařízením je u *třídy A* povoleno pouze po úspěšné komunikaci v opačném směru, tedy z koncového zařízení směrem k hraniční bráně.

Zařízení *třídy B* obohacují *třídu A* o další metodu komunikace v příchozím směru. Koncová zařízení v této kategorii jsou synchronizována pomocí periodické signalizace vysílané hraničními branami, což umožňuje koncovým zařízením přijmout data i bez předchozí úspěšné komunikace v odchozím směru. Nutnost synchronizace neumožňuje modemu využít režimu spánku a zařízení *třídy B*, tak mají vyšší spotřebu elektrické energie než zařízení *třídy A*.

Třetí kategorií jsou zařízení *třídy C*, které poslouchají na komunikačním kanále neustále a jsou schopné přijmout data kdykoliv. Tato zařízení jsou energeticky nejnáročnější.

Všechny tři kategorie musí také dodržovat regulační normy o maximální době vysílání. Pro evropské pásmo 868 MHz je např. definováno maximální využití pásma 1 % pro každé koncové zařízení. Regulace využití pásma a *ALOHA* přístup k síti limituje maximální počet koncových zařízení, které je možné připojit k jedné hraniční bráně. Závislost přenosových rychlostí na počtu koncových zařízení připojených k hraniční bráně můžeme pozorovat v Tab. 2.1. Zobrazeny jsou hodnoty pro 3 kanály a datový obsah zprávy 30 bytů.

Tab. 2.1: Maximální propustnost LoRaWAN sítě.

počet koncových zařízení	500	1000	5000
maximální propustnost (paketů / h)	117	53	10
maximální propustnost (bytů / h)	3510	1590	300

## 2.3 Možnosti využití

LoRaWAN patří mezi rozšířenější technologie v oblasti LPWAN sítí. Velký tlak na nízkou cenu, spotřebu a vysoký dosah komunikujících zařízení však do standardu přinesl řadu kompromisů. Mezi hlavní limitace LoRaWAN sítí patří nízká přenosová rychlost, vysoké zpoždění rychle rostoucí s přibývajícím množstvím koncových zařízení a omezený počet koncových zařízení v síti převážně způsobený přístupovým mechanismem prostá ALOHA. Silné stránky a limitace LoRaWAN technologie je třeba brát v úvahu při diskuzi vhodných využití v následujících sekcích.

### 2.3.1 Odečet dat v reálném čase

Pro tuto kategorii je při použití LoRaWAN technologie nutné klasifikovat nároky na zpoždění pro danou aplikaci. Mechanismus prostá ALOHA, který standard využívá, se vyznačuje rychle rostoucím zpožděním v případě častých kolizí. Počet kolizí stoupá s počtem koncových zařízení v síti a délkou odesílaných dat. Nedeterministický charakter přístupové metody prostá ALOHA tak není vhodný pro kritické aplikace s požadavky na minimální odezvu, tedy např. řízení výrobních linek.

LoRaWAN v této kategorii však najde uplatnění např. v zemědělství. Senzorické jednotky umožňují sbírat cenná data jako vlhkost a salinitu půdy, intenzitu slunečního záření, rychlost větru a teplotu, která mohou být v reálném čase odesílána k vyhodnocení. Z dat je pak možné reagovat téměř v reálném čase upravením podmínek pro pěstované plodiny nebo chovanou zvěř.

Podobné požadavky klade také sběr environmentálních dat. Využití LPWAN sítí umožňuje výrazně hustější pokrytí vybraného území meteorologickými stanicemi. Relativně levné senzory pro měření základních meteorologických veličin jako teplota, vlhkost, tlak vzduchu a nízká spotřeba elektrické energie zařízení komunikujících v LoRaWAN síti umožňují masivní nasazení takových jednotek. Vhodnost nasazení výše zmíněného řešení je diskutováno v kapitole 3, kde je také popsán vývoj a testování podobného zařízení.

Další aplikací, která vyhovuje limitům zmíněným na začátku této sekce je např. odečet výšky vodní hladiny. V oblastech často postihovaných záplavami je vhodné nasadit řešení, které je schopné informovat o záplavové hrozbě co nejdříve, aby bylo možné minimalizovat škody.

### 2.3.2 Chytrá města

Ačkoliv by tuto kategorii šlo zařadit mezi odečet dat v reálném čase, viz 2.3.1, velká medializace využití LPWAN sítí pro automatizaci měst nutí pro toto využití použít

samostatnou kategorii. Nejčastěji diskutovanými aplikacemi jsou *chytré parkování* a *chytré odpadkové koše*.

V první aplikaci LoRaWAN koncová zařízení sledují obsazenost parkovacích míst (nejčastěji za použití ultrazvukových distančních senzorů) a informace jsou v reálném čase poskytovány řidičům hledajícím volné parkovací místo.

Druhá aplikace využívá senzorů uvnitř odpadkových košů, které o své zaplněnosti informují odpadkovou službu, a ta je díky informacím schopná efektivněji provádět svoz odpadků.

## 3 REALIZACE LORAWAN ZAŘÍZENÍ

Praktická část této práce se zabývá návrhem bateriově napájené meteorologické jednotky, která snímá veličiny okolního prostředí a pomocí LoRaWAN sítě je odesílá k dalšímu zpracování na aplikační server. Pro účely přehledného popisu řešení je tato kapitola rozdělena do čtyř sekcí. V první sekci je popsáno samotné zařízení sbírající meteorologická data okolí, použité senzory, jejich komunikační protokoly a LoRaWAN komunikační modul. Druhá sekce se zaměřuje na síťový server LORIoT, který slouží jako sběrný bod všech LoRaWAN zařízení připojených do sítě. Třetí sekce popisuje meteorologický aplikační server, který data z meteorologické jednotky nejprve získává z síťového serveru a následně je ukládá a třídí. V poslední sekci se práce zaměřuje na zobrazení a interpretaci dat z meteorologické jednotky.

Rozdělení na čtyři sekce zároveň kopíruje celkovou architekturu systému, která je naznačena na obr. 3.1. Prvky jsou odlišeny barevně podle sekcí, ve kterých budou popsány a LoRa komunikace je znázorněna přerušovanou čarou.

### 3.1 Meteorologická jednotka

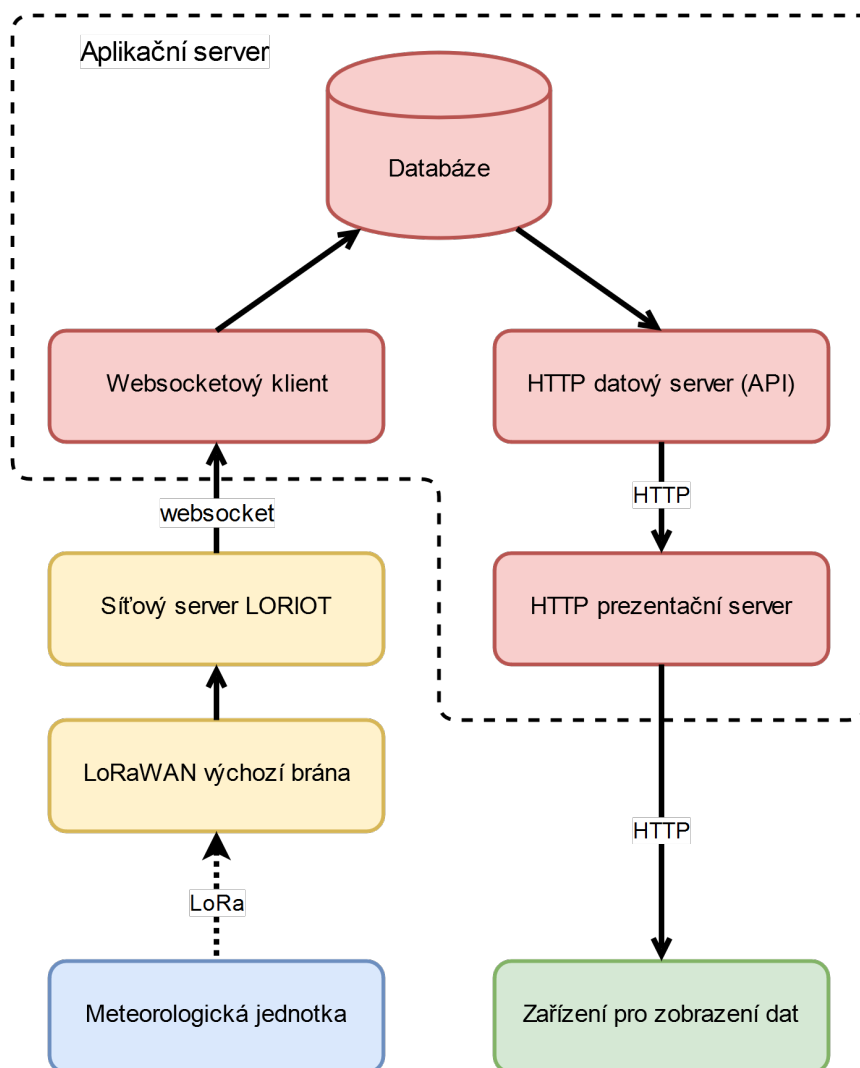
Meteorologická jednotka je zařízení umístěné do míst, kde je potřeba odečítat meteorologické fyzikální veličiny. Jako stěžejní byly vybrány tyto veličiny:

- Teplota okolního prostředí.
- Relativní vlhkost vzduchu.
- Atmosferický tlak.

Jednotka se skládá z MCU (jednočipový počítač - Microcontroller Unit) vývojové desky Arduino UNO, LoRaWAN SoM (systém na modulu - System on Module) založeném na SoC od společnosti Semtech nesoucím označení SX1276 a z jednotlivých senzorů, které slouží k převodu meteorologických fyzikálních veličin na veličiny elektrické.

#### 3.1.1 MCU Arduino

Arduino je název produktové řady stejnojmenné společnosti, která zahrnuje jednodeskové počítače založené na osmibitových MCU společnosti Atmel. (V posledních několika letech se rozšiřují také verze s 32-bitovým MCU.) Jednodeskové počítače vznikly jako jednoduchá platforma pro rychlé vytvoření prototypů, výuku teorie digitálních, ale i analogových obvodů a převážně jako metoda, jak k elektrotechnice přivést širší veřejnost. Myšlenka se velmi rychle stala populární, mimo jiné také z důvodu uvolnění elektrotechnických schémat a zdrojových kódů jako OSHW (otevřený hardware - Open-source Hardware) a OSS (otevřený software - Open-source



Obr. 3.1: Architektura systému pro měření meteorologických veličin.

Software). V současnosti je platforma Arduino dostatečně vyspělá, pro přímé využití v komerčních produktech. Nejdůležitějšími částmi desky Arduino jsou paměť, procesor (MCU) a periferní obvody.

Arduino UNO využívá MCU Atmel ATmega328, který obsahuje tři typy paměti. Paměť Flash slouží k uchování programového kódu, který má procesor vykonávat. Paměť SRAM (statická paměť - Static Random-Access Memory) je paměť, kde programový kód dynamicky alokuje paměť a ukládá proměnné. Posledním typem paměti je EEPROM (Electrically Erasable Programmable Read-Only Memory), která slouží pro dlouhodobější uchování dat, v meteorologické stanici lze paměť EEPROM využít například k místnímu uložení naměřených dat. Paměti Flash a EEPROM jsou nevolatilní, což znamená, že uchovávají uložená data i při odpojení napájení. Paměť SRAM je naopak volatilní, tedy podobně jako paměť DRAM (dynamická paměť

- Dynamic Random-Access Memory) v moderních počítačích není schopná udržet data po odpojení napájení. Přehled velikostí jednotlivých typů paměti je uveden v tabulce 3.1. Při vytváření programového kódu je nutné dbát na to, aby velikost přeloženého programu nepřesáhla velikost Flash paměti, a protože je na paměti Flash uložen také programový zavadač, který slouží k usnadnění programování procesoru, není možné pro vlastní programový kód využít plných 32 kB Flash paměti. Velikost SRAM paměti může pro programátora taktéž představovat určitá omezení a to obzvláště při programování uživatelského rozhraní, kde se předpokládá hojné využití textových řetězců, které jsou náročné na velikost alokované paměti.

Tab. 3.1: Velikost pamětí v MCU ATmega328.

Typ paměti	Velikost
Flash	32 kB
SRAM	2 kB
EEPROM	1 kB

Procesor ATmega328 standardně operuje na vnitřní frekvenci 16 MHz a jedná se o osmibitový procesor. Osmibitový procesor se vyznačuje tím, že všechny jeho instrukce umí jako vstup přijmout číslo o maximální velikosti 8 bitů. Pokud je třeba operovat s čísly většími, například sčítání dvou 16-bitových čísel, nelze tuto operaci provést v jedné instrukci a je nutné vykonat instrukcí několik. Osmibitový procesor tedy není pomalejší ve vykonávání instrukcí oproti jeho 16-bitové variantě, nicméně při operacích, kde vstupní data přesahují velikost 8 bitů je mnohem méně efektivní.

MCU kromě procesorového jádra obsahuje také řadu periferních obvodů. GPIO (všeobecně použitelný vstup/výstup - General Purpose Input/Output) porty slouží k ovládání a sběru informací z obvodů mimo samotné MCU. Periferie dále zahrnují komunikační obvody s fixní funkcionalitou podporující komunikační protokoly SPI (Serial Peripheral Interface) a I<sup>2</sup>C (Inter-Integrated Circuit). V neposlední řadě ATmega328 obsahuje A/D převodník s šesti multiplexovanými vstupy, které zprostředkovávají možnost pracovat s externími analogovými signály.

Jednodeskový počítač kromě samotného MCU obsahuje krystal, který je nezbytný pro udržení stabilní frekvence vnitřního časového signálu MCU a vykonávání jeho instrukcí, napájecí obvod a převodník z USB (Universal Serial Bus) na sériové rozhraní UART (asynchronní sériové rozhraní - Universal Asynchronous Receiver-Transmitter), který umožňuje platformě Arduino komunikovat s moderními počítači.



### 3.1.2 LoRaWAN modul SX1276

Modul od společnosti Semtech je modem plně kompatibilní s LoRaWAN specifikací koncového zařízení typu A. To znamená, že meteorologická jednotka bude mít nízké nároky na spotřebu elektrické energie, ale nebude moct přijmout žádná data mimo krátký interval ihned po odeslání dat na LoRaWAN výchozí bránu. Pro zvolenou aplikaci to však není limitující, jelikož všechna komunikace probíhá ve směru z koncových zařízení na síťový server, viz obr. 3.1.

Modul SX1276 neobsahuje kompletní software nutný pro realizaci LoRaWAN komunikace jako některé jiné moduly, které podporují ovládání pouze přes speciální AT příkazy, s významy jednotlivých AT příkazů obvykle uvedených v dokumentaci modulu. Modul SX1276 proto vyžaduje specializovaný software na straně MCU, který řídí komunikaci, stará se o znovuodeslání nedoručených zpráv a obecně o to, aby komunikace odpovídala standardu LoRaWAN. Se samotným MCU modul komunikuje přes rozhraní SPI, které je popsáno níže v podsekcí 3.1.4.

### 3.1.3 Senzory meteorologických veličin

Pro účely meteorologické jednotky byly na základě výše vybraných požadavků na měřené veličiny zvoleny dva senzory. Senzor atmosferického tlaku a teploty BMP180 vyráběný společností Bosch a senzor teploty a relativní vlhkosti vzduchu SHT30 od společnosti Sensirion. Oba senzory mají výhodu již integrovaných A/D převodníkům a komunikují tedy s MCU přes digitální rozhraní. Díky tomu dosahují vyšší přesnosti a jejich obsluha probíhá za použití vyšší míry softwarové abstrakce, což snižuje náročnost implementace. Jak senzor SHT30, tak BMP180 jsou realizovány formou integrovaného obvodu a zapouzdřeny do pouzder SMD (součástka pro povrchovou montáž - Surface-Mount Device) o velikostech 3,6 x 3,8 mm a 2,5 x 2,5 mm, připravené k osazení na desku plošných spojů.

Senzor SHT30 dle specifikací dosahuje typické odchylky pro teplotu 0.2 °C a 2 % relativní vlhkosti [20]. Pro senzor BMP180 je absolutní odchylka měřeného atmosferického tlaku uvedena 2 hPa [21]. Uvedené přesnosti senzorů lze považovat za nadprůměrné a dostačující pro zvolenou aplikaci.

Oba senzory pro komunikaci s MCU využívají rozhraní I<sup>2</sup>C, které je popsáno v podsekcí 3.1.4.

### 3.1.4 Rozhraní SPI a I<sup>2</sup>C

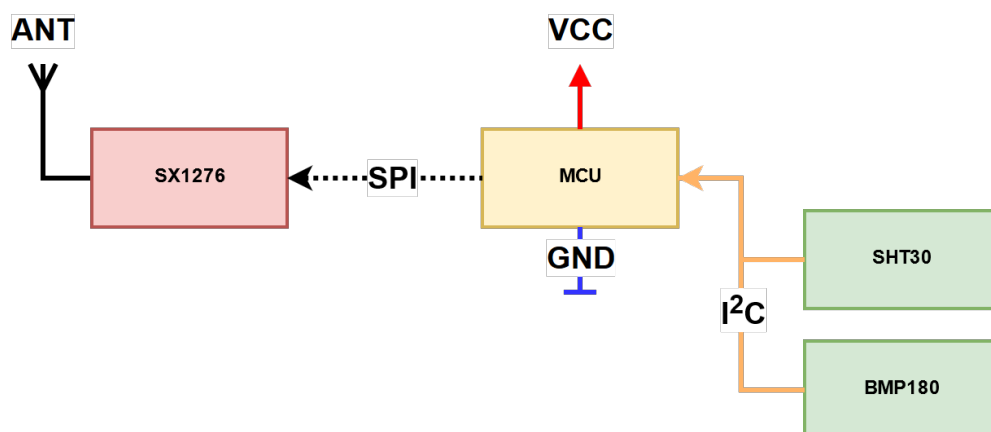
Obě rozhraní představují standardizovaný způsob komunikace mezi několika MCU nebo mezi MCU a periferními komponentami. K přenosu dat využívají linkové kódy, což znamená, že přenášejí v základním pásmu bez použití modulace.

SPI využívá třívodičové sběrnice pro komunikaci a  $n - 1$  vodičů k adresaci jednotlivých zařízení. Na sběrnici mohou v jednu chvíli komunikovat vždy jen dvě zařízení. Z toho jedno musí zastávat funkci *master* a druhé *slave*. Funkcionalita vodičů společné sběrnice je následující: SCLK (hodinový signál - Serial Clock) vodič šíří hodinový signál, MOSI (master výstup, slave vstup - Master Output Slave Input) vodič slouží ke komunikaci ve směru *master*  $>$  *slave*, MISO (master vstup, slave výstup - Master Output Slave Input) naopak ke komunikaci *slave*  $>$  *master*. Vodiče sloužící k adresaci zařízení se označují jako SS (Slave Select).

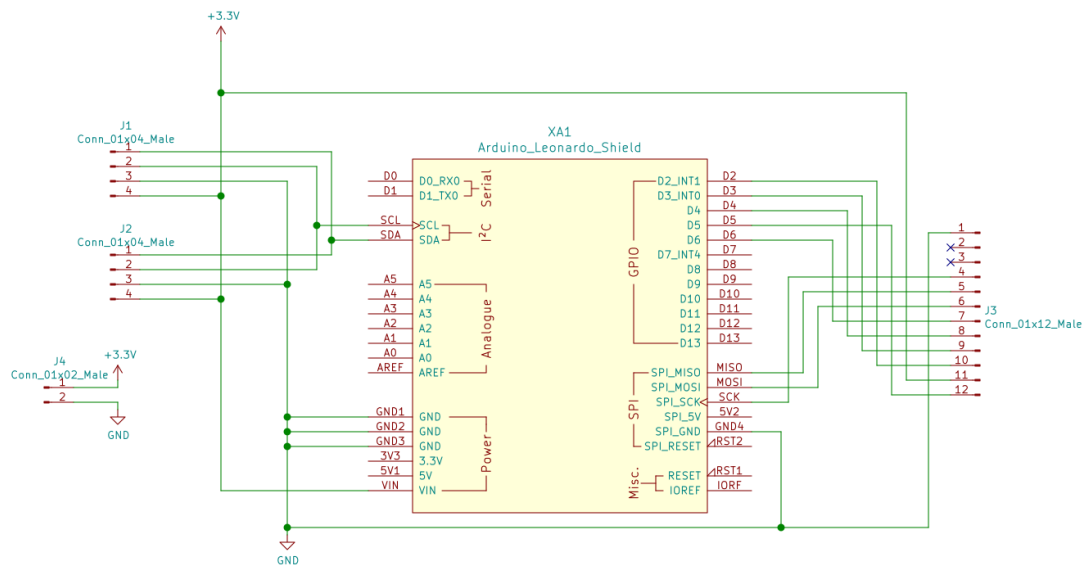
I<sup>2</sup>C taktéž využívá sdílené sběrnice, která je ale dvouvodičová. Sběrnice se skládá z vodiče s hodinovým signálem SCL (hodinový signál - Serial Clock Line) a datového vodiče SDA (datový signál - Serial Data Line), který je na rozdíl od SPI využíván v duplexním přenosovém režimu. Hlavní rozdíl mezi I<sup>2</sup>C a SPI je ale v přístupu, jaký volí k adresaci zařízení. Zatímco SPI potřebuje pro každé další zařízení vyhrazený SS vodič, I<sup>2</sup>C zařízení adresuje přes SDA vodič pomocí unikátní adresy. Využíván je obvykle sedmibitový adresní prostor, který teoreticky znamená omezení počtu připojených zařízení na 127, ale v praxi je tento limit většinou více než dostačující.

### 3.1.5 Implementace koncového zařízení

Arduino MCU, LoRaWAN modul SX1276 a senzory byly za pomoci výše popsaných rozhraní propojeny podle blokového schématu na obr. 3.2. Propojení je realizováno pomocí desky plošných spojů, jehož schéma je uvedeno na obr. 3.3 a návrh desky na obr. 3.4. Finální podoba vyhotovené desky osazené dutinkovými lištami a jedním senzorem je uvedena na obr. 3.5.



Obr. 3.2: Blokové schéma meteorologické jednotky.

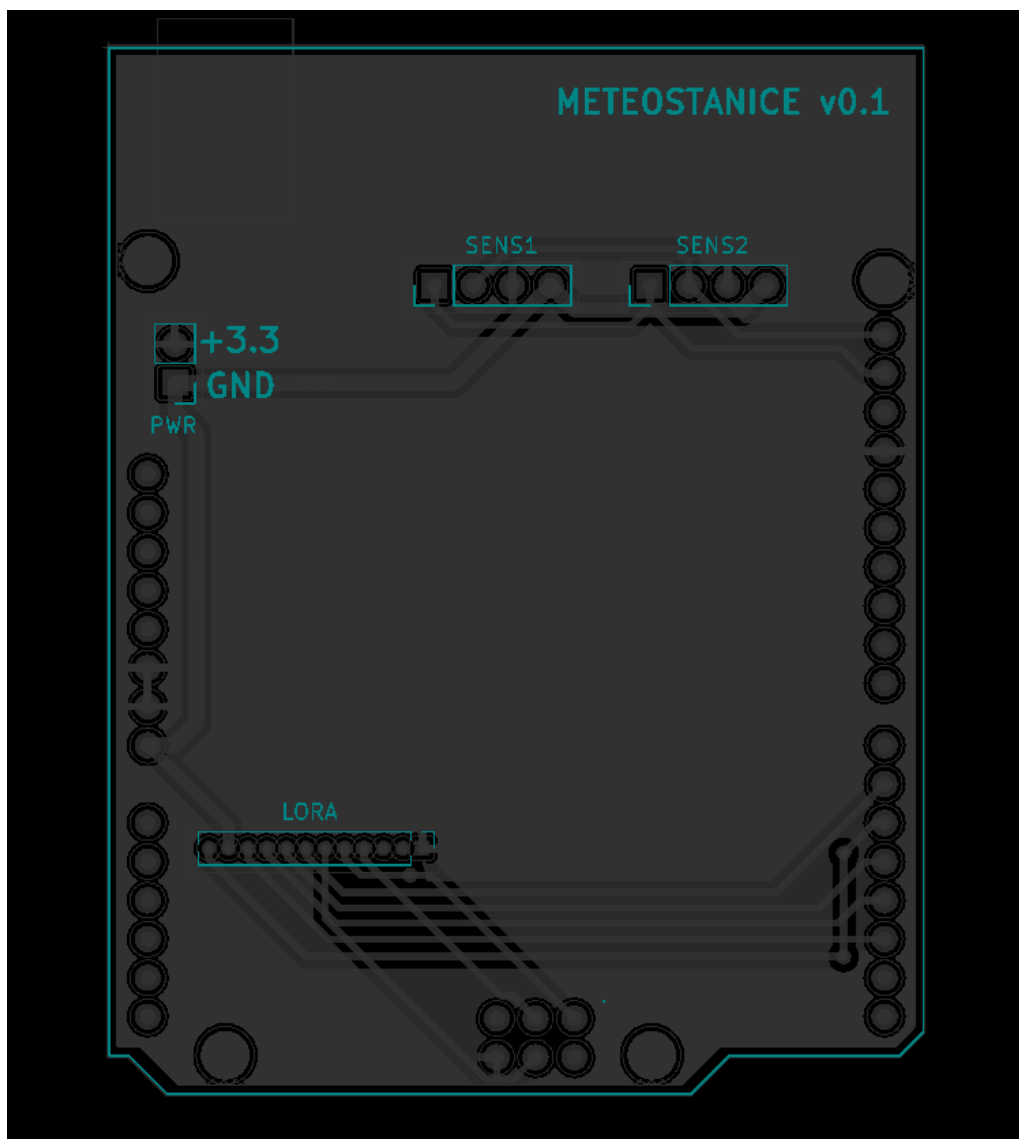


Obr. 3.3: Elektronické schéma zapojení součástek jednotky.

Deska plošných spojů byla navržena s využitím návrhového programu KiCAD. Schéma vychází ze záměru udělat platformu meteorologické stanice co nejvíce flexibilní. I<sup>2</sup>C sběrnice pro senzory byla vyvedena na standardní dutinkovou lištu s rozpětím 2,54 mm, aby bylo možné vybrané senzory v budoucnosti vyměnit za jiné. Napájení je taktéž realizováno pinem pro připojení externího napájecího modulu, který může obsahovat obvod podporující napájení z baterie, zároveň je však možné stanici připojit přímo k laboratornímu zdroji.

Jako napájecí napětí bylo zvoleno napětí 3,3 V. Ačkoliv toto napájecí napětí není výchozím napětím (5 V) na kterém deska Arduino pracuje, tak se stále nachází v rozmezí, při kterém je deska schopna bez potíží operovat a toto rozhodnutí přináší v návrhu řadu výhod. Zaprvé snížení napájecího napětí znamená, že není třeba využít převodníku úrovní pro komunikaci s LoRaWAN modulem SX1276 a senzorem BMP180. Obě zařízení vyžadují digitální komunikaci s logickými úrovněmi nepřesahující napětí 3,3 V a Arduino své napájecí napětí využívá jako referenční napětí pro komunikační piny. Zadruhé nižší napětí vede ke snížení příkonu MCU, což je u bateriově napájeného zařízení žádoucí.

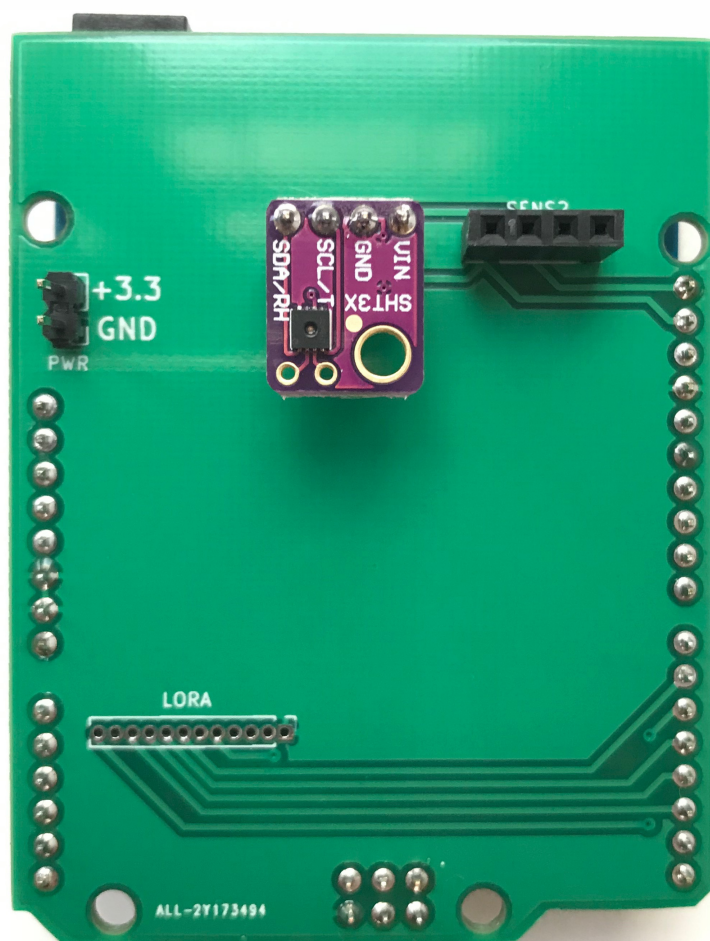
Jednotlivé komponenty byly na desce plošných spojů rozmístěny tak, aby jeho realizace byla snadná a také s ohledem na potřeby komponent. Napájení bylo umístěno na okraj desky pro snadné připojení konektoru. Na kraj desky byl také umístěn LoRaWAN modul převážně proto, aby byla jeho anténa mimo vrstvy mědi a ty nepůsobily jako stínění bránící průchodu signálu. Senzory pak byly umístěny blízko



Obr. 3.4: Návrh desky plošných spojů.

vývodů I<sup>2</sup>C komunikace a tak, aby si vzájemně nepřekážely. Pro realizaci desky bylo nutné využít dvě vrstvy mědi s prokovením.

K ovládání modulu byla použita knihovna vyvinutá společností IBM (International Business Machines Corporation) s názvem *Arduino-LMIC* dostupná z [22]. Pro přihlášení do sítě je nutné koncovému zařízení nastavit několik parametrů. Nejprve je nutné v mikrokontroleru nastavit správné přiřazení funkcí jednotlivým pinům, dále je nutné určit metodu autentizace, kterou bude zařízení využívat. V tomto případě se jedná o metodu ABP (Activation-by-Personalisation), což znamená, že se musí nastavit *NSK* (*network session key*) a *ASK* (*application session key*). Tyto klíče zajistí autentizaci koncového zařízení v LoRaWAN síti. Posledním parametrem, který je nutné zařízení specifikovat je jeho adresa. Adresa musí být v celé síti jedinečná.



Obr. 3.5: Vyhotovená deska plošných spojů.

Nastavení těchto a dalších méně důležitých parametrů je vidět v příloze A.1.

Mikrokontroler komunikuje se senzory přes I<sup>2</sup>C sběrnici a za tímto účelem je využívána oficiální Arduino knihovna Wire [23]. Pro samotnou komunikaci je však využito ještě i dalších knihoven - knihovna *arduino-sht* od vydavatele Sensirion dostupná z [24] a knihovna *Adafruit-BMP085-Library* od vydavatele Adafruit dostupná z [25], které zprostředkovávají vyšší vrstvy komunikace se senzory - např. posílají zprávy s požadavkem na čtení hodnoty a kontrolují cyklický redundantní kód odpovědi.

Po odečtení dat je nutné data ze senzorů připravit do podoby vhodné k odeslání přes LoRaWAN síť. Jednou z možností je data zpracovat do datového formátu JSON (JavaScript Object Notation). JSON je průmyslový standard a jeho zpracování má velkou podporu napříč programovacími jazyky. Výhodou je, že při vývoji programů

zpracovávající tento formát není nutné se starat o zpracování příchozích dat. Velkou nevýhodou standardu je ale fakt, že dodržením syntaxe JSON formátu délka zprávy vzroste o 40 %, viz tabulka 3.2. Proto byl zvolen vlastní úsporný formát dat ve tvaru

$$t : \{hodnota\ teploty\}, p : \{hodnota\ tlaku\}, h : \{hodnota\ vlhkosti\}.$$

V tomto tvaru odesílá meteorologická jednotka svá data na LoRa hraniční brány, které je přeposílají na síťový server LORIoT. Velikost zprávy lze ještě dále optimalizovat například odstraněním desetinných teček, odstraněním nadbytečných nul v desetinné části hodnot a odstraněním identifikátorů položek. Oproti použitému formátu tak lze odstranit dalších 40 % velikosti zprávy. Od tohoto formátu však bylo upuštěno pro zhoršenou čitelnost.

Tab. 3.2: Porovnání JSON a vlastního formátu předávaných dat.

Notace zprávy	Formátovaná zpráva	Velikost zprávy
JSON	{"t":22.50,"p":98645,"h":54}	28 B
vlastní	t:22.50,p:98645,h:54	20 B
optimalizováno	225,98645,54	12 B

### 3.1.6 Režim snížené spotřeby

U aplikace jako je sběr meteorologických dat lze předpokládat využití bateriového napájení pro sběrné stanice. Provoz mikrokontroleru a radiového vysílače je však energeticky náročná činnost. Proto bylo v rámci práce zkoumány možnosti snížení spotřeby sběrného zařízení a výsledky jsou popsány v této sekci.

Energetický odběr LoRaWAN modulu není možné přímo ovlivnit. Modul automaticky využívá režim snížené spotřeby (*sleep state*) mimo čas, kdy zrovna vysílá nebo přijímá a v tomto režimu je jeho odběr maximálně 200 nA. Největší možnost omezení energetické spotřeby je tedy na straně MCU.

Vývojová platforma Arduino je ideální k urychlení vývoje, ale není uzpůsobena pro bateriově napájené zařízení. Tyto vývojové desky obsahují několik obvodů, které snižují energetickou efektivitu zařízení a jejichž vyřazení může značně prodloužit dobu provozu při využití bateriového napájení. Proudový odběr vývojové desky Arduino UNO při napájecím napětí 5 V byl naměřen 50 mA. Po odstranění signalizačních diod, regulátoru napětí, dalších nadbytečných obvodů a snížení napájecího napětí na 3,3 V se proudový odběr snížil na 4,5 mA.

Dalšího snížení lze dosáhnout využitím režimů spánku, které výrazně snižují spotřebu mikrokontroleru v době nečinnosti. MCU ATmega328 podporuje několik

režimů spánku, avšak ne všechny jsou vhodné. Některé velmi úsporné režimy vyžadují pro následné probuzení vnější přerušení na předem zvoleném vstupu MCU a to není pro realizaci meteorologické stanice vhodné. Proto byl vybrán spánkový režim takový, který umožňuje následné probuzení po předem určeném čase. Zároveň byly použity další metody snížení spotřeby popsané v [26] a dostupné v knihovně pro Arduino platformu vydané pod názvem LowPower [27]. Celkový energetický odběr v režimu spánku se snížil na  $8,9 \mu\text{A}$ .

Určením dalších podmínek provozu můžeme odhadnout životnost bateriově napájeného zařízení. Bylo vyzkoušeno, že LoRaWAN modul a obslužný mikrokontroler potřebují k odeslání zprávy přibližně 0,5 s a odběr je v tomto časovém intervalu průměrně 50 mA. Pro výpočet výdrže baterie bylo předpokládáno odesílání zpráv každých 15 minut, využití 3 alkalických baterií s celkovou kapacitou 9 Wh a účinnosti regulátoru napětí 90 %. Na základě těchto vstupních dat a vztahu 3.1 byla výdrž baterií odhadnuta na 4 roky, přičemž po dosazení je zřejmé, že největší faktor je samovybíjecí proud baterie.

$$t = \frac{\text{kapacita baterie}}{P_{SYS} + P_{SD}} \cdot \frac{1}{\eta}, \quad (3.1)$$

kde:

$P_{SYS}$ : je průměrný příkon stanice.

$P_{SD}$ : je průměrný výkon ztracený samovybíjením baterie.

$\eta$ : je účinnost napěťového regulátoru za baterií.

## 3.2 Síťový server LORIOT

Síťový server LORIOT slouží jako sběrný bod pro všechny nasazené meteorologické jednotky. LORIOT je komerční technologie plnící roli síťového serveru. Server vytváří nad sítí logické celky, autentizuje koncová zařízení, komunikuje s LoRaWAN hraničními branami a sesbíraná data přeposílá dále k zpracování zákazníkem. Server je zároveň připraven poskytovat LoRaWAN konektivitu jako službu, což představuje podobný obchodní model, jako dnes poskytují mobilní operátoři na svých buňkových sítích.

Preferovanou metodou komunikace LORIOT serveru se zákazníkem je přes technologii WebSocket, což je komunikační protokol pracující na bázi HTTP (Hypertext Transfer Protocol), ale po sestavení TCP (Transmission Control Protocol) spojení jej udržuje otevřené a umožňuje obousměrnou komunikaci s nižší režii, než v případě HTTP.

Pro sledování komunikace na WebSocketu byl připraven soubor, který je možné otevřít v internetovém prohlížeči. Soubor obsahuje kód programovacího jazyka Ja-

JavaScript z důvodu jeho jednoduché implementaci do prohlížeče internetových stránek. Software k sledování komunikace je velmi jednoduchý. Internetový prohlížeč se pokusí navázat WebSocket spojení na adresu LORIOT síťového serveru a autentizuje se předem připraveným klíčem. Pokud spojení neselže, internetový prohlížeč naslouchá na WebSocketovém spojení a čeká na příchozí zprávy. Pokud přijde zpráva, JavaScriptový kód zpracuje příchozí data a naplní vhodná pole. Zobrazuje se teplota, atmosferický tlak a SNR (odstup signálu od šumu - Signal to Noise Ratio) přijímaného signálu z koncového zařízení. Konkrétní softwarovou implementaci je možné najít v příloze C.1. Ve skutečnosti však přes WebSocket s LORIOT serverem komunikuje aplikační server, který je popsán v sekci 3.3.

### 3.3 Aplikační server

Účel aplikačního serveru je sbírat data ze síťového serveru LORIOT, ukládat je do databáze a připravit do formy vhodné pro prezentování uživateli. Aplikační server zahrnuje čtyři části: WebSocketový klient, databáze, datový server a prezentační server. Každá část plní specifickou roli a části mezi sebou komunikují pomocí různých protokolů. Fyzicky se však všechny části nachází na jednom zařízení. Jako programovací jazyk pro vytvoření aplikačního serveru byl zvolen Python. V následujících podsekcích jsou popsány části aplikačního serveru.

#### 3.3.1 WebSocketový klient

WebSocketový klient je jednoduchý program, který navazuje spojení na LORIOT server pomocí WebSocket protokolu a přijímá zprávy, které síťový server LORIOT přeposílá z meteorologických stanic. WebSocketový klient přijaté zprávy zpracuje, upraví do požadovaného formátu a uloží do databáze ke správné stanici.

Samotný program kontroluje jestli je spojení aktivní a v případě ztráty spojení nebo při přesáhnutí maximální doby čekání na zprávu se pokusí spojení obnovit, aby nedocházelo k ztrátě dat.

#### 3.3.2 Databáze

Databáze plní roli dlouhodobého úložiště strukturovaných meteorologických dat. Za tímto účelem byl zvolen open-source software MariaDB. MariaDB vychází ze známějšího databázového softwaru MySQL a je jeho téměř totožnou kopií, avšak na rozdíl od MySQL splňuje všechny požadavky open-source softwaru. Jedná se o SQL (strukturovaný dotazovací jazyk - Structured Query Language) databázi, což znamená, že veškerá komunikace s databází musí odpovídat syntaxi SQL jazyka.



SQL je jazyk vyvinutý přímo za účelem využití v databázích a je vhodný pro popis strukturovaných dat.

Každou SQL databázi lze popsat pomocí databázového schématu, které uloženým datům dává přesnou strukturu a definuje mezi datovými strukturami vzájemné vztahy. Databázové schéma databáze využitě v meteorologickém aplikačním serveru je základní a výpis schématu odpovídá následujícímu výpisu:

```
CREATE TABLE stanice (  
    id INTEGER NOT NULL,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE zaznam (  
    id INTEGER NOT NULL,  
    id_stanice INTEGER,  
    timestamp DATETIME,  
    data VARCHAR(250),  
    PRIMARY KEY (id),  
    FOREIGN KEY(id_stanice) REFERENCES stanice (id)  
);
```

Databáze obvykle pracují s architekturou klient-server. Serverem je samotný software databáze a klientskou část představuje aplikační kód, který do databáze data ukládá nebo data z databáze získává. Klienti pro manipulaci dat využívají SQL jazyk. Požadavek na výpis všech meteorologických záznamů stanice s identifikátorem  $id=1$  lze získat správným dotazem 3.2, který databázový server nejprve interpretuje, na jeho základě získá data z úložiště a nakonec je vrátí v oběma stranám známém formátu.

$$SELECT * FROM zaznam WHERE id\_stanice = 1; \quad (3.2)$$

### 3.3.3 Datový server

Data uložená v databázi jsou dostupná jen v rámci aplikačního serveru. Přístup k databázi sice lze umožnit přes síť, ale toto chování není žádoucí z důvodu bezpečnosti a výkonnosti. Proto byl vytvořen datový server, který plní právě záměr zpřístupnit naměřená data dalším aplikacím a uživatelům. Datový server je HTTP server, který zpracovává HTTP dotazy, vyhledává v databázi a vrací odpovědi v JSON formátu. Pro naprogramování byl vybrán jazyk Python a knihovny Flask a SQLAlchemy, které vytvoření serveru usnadňují.

Pomocí HTTP serveru bylo vytvořeno tzv. API (aplikační rozhraní - Application Programming Interface), které slouží jako standardizovaný přístupový bod pro další aplikace, které chtějí s meteorologickými daty pracovat. API neslouží k zobrazení dat v čitelné podobě koncovým uživatelům, ale k vytvoření vhodného prostředí pro aplikace třetích stran, které chtějí meteorologická data pro své účely získat. Aplikační rozhraní tedy nechává samotnou prezentaci dat na dalších aplikacích a soustředí se na standardizovaný způsob přístupu k záznamům uloženým v databázi, ke které v případě meteorologických dat přistupuje v režimu pouze pro čtení. Záměrem každého aplikačního rozhraní by mělo být, aby docházelo ke změnám ve formátu dotazů a vracených dat co nejméně, protože každá taková změna může způsobit omezení funkčnosti aplikací, které rozhraní využívají. API meteorologického datového serveru implementuje pouze jedinou metodu

*/nodeapi/start/ < datum\_start > /end/ < datum\_konec > .*

Tato metoda přijímá jako vstupní parametry dva dny ve formátu *YYYY-MM-DD*, které definují časový interval, pro který chceme výpis meteorologických dat. Odpověď na dotaz odpovídající této metodě je obsažena ve výpisu:

```
[
  [
    "2018-01-19T06:50:24",
    {
      "pres": 96322,
      "temp": 21.1
    }
  ],
  [
    "2018-01-19T06:46:20",
    {
      "pres": 96313,
      "temp": 21.1
    }
  ]
]
```

Aby datový server mohl získat databázové záznamy a odpovídat na dotazy uživatelů, musí s ostatními stranami komunikovat standardními protokoly. S využitím těchto protokolů nám pomáhají knihovny programovacího jazyka.

Knihovna Flask zajišťuje správné zpracování HTTP dotazů, které na server posílají klientské aplikace případně rovnou uživatelé. Knihovna se stará o správné nastavení všech hlaviček protokolu, nastavuje délku vrácené odpovědi a obecně zaručuje,

že náš server bude odpovídat oficiální dokumentaci protokolu HTTP. Nejdůležitější soubor využívaný touto knihovnou obsahuje tzv. směrovač dotazů. V tomto souboru jsou definovány implementované metody aplikačního rozhraní a jejich URL (jednotná adresa zdroje - Uniform Resource Locator) cesty. Každé metodě je přiřazena programová funkce, která má být použita ke zpracování dotazu a vytvoření odpovědi. Obsah souboru se směrovačem dotazů je uveden v příloze B.1.

Knihovna SQLAlchemy slouží k zjednodušení komunikace datového serveru s databází. jedná se o ORM (objektově relační mapování - Object-relational mapping) nástroj, který figuruje jako vrstva mezi aplikačním kódem a databází. ORM nástroj má za úkol zavést určitou míru abstrakce nad databázovými tabulkami. Tabulky nástroj převádí na třídy objektově orientovaného kódu a jednotlivé záznamy v tabulkách pak na objekty těchto tříd. Mapování tabulek na třídy se ukládá v souboru s definicí modelů, jehož výpis lze najít v příloze B.2.

### 3.4 Interpretace zasílaných dat

Záznamy o meteorologických veličinách poskytované datovým serverem jsou pro koncového uživatele nečitelné a je tedy žádoucí je převést do přehledné formy. Jelikož se záznamy skládají z hodnot meteorologické veličiny a časového údaje byl jako vhodná forma zobrazení zvolen graf závislosti meteorologické veličiny na čase. Za účelem vykreslení grafů byla připravena prezentační aplikace naprogramovaná v programovacím jazyce JavaScript.

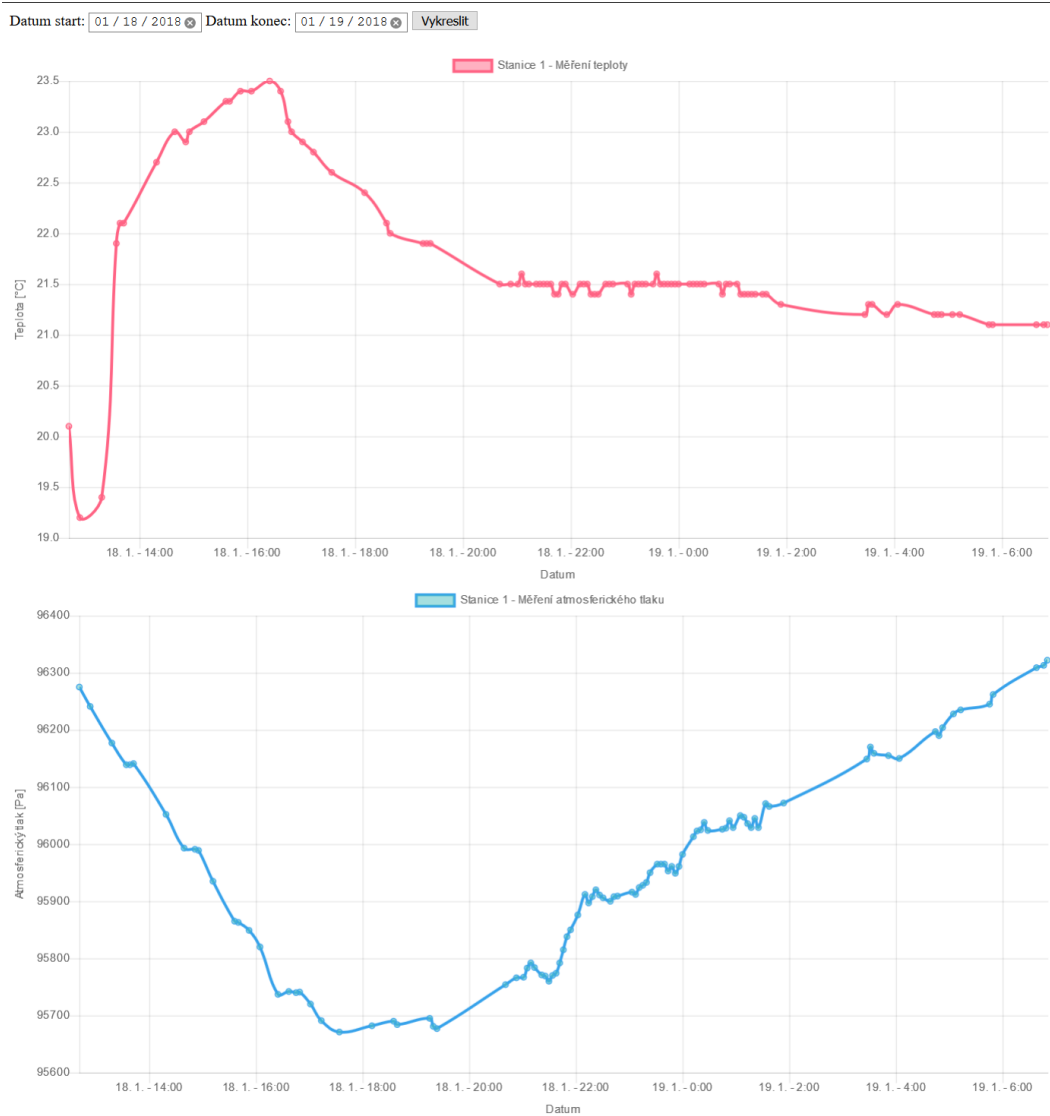
Prezentační aplikace komunikuje s API datového serveru a obdržaná data vykresluje formou přehledných grafů. Komunikace s API probíhá asynchronně a neblokuje tedy uživatele od používání aplikace ani v případě velmi dlouhé odezvy datového serveru.

Aplikace jako vstup od uživatele přijímá datum začátku a datum konce, které vymezují časový interval pro který si uživatel přeje vykreslit graf meteorologických záznamů. Data jsou předávána formou kolonek, které po stisknutí nabízí dialog s výběrem data. Po vybrání vstupních dat a stisknutí tlačítka vykreslit aplikace odesílá dotaz na datový server a na základě vrácené odpovědi vykreslí grafy. Zároveň si hodnoty dat vymezujících interval uloží ve formě URL parametrů. Díky tomu mohou uživatelé mezi sebou sdílet vykreslené grafy pomocí URL adresy a není třeba sdílet fotku.

O samotné vykreslení grafů se stará knihovna Chart.js, která byla vybrána pro svou jednoduchost, a protože se jedná o zdarma využitelný software, který je vydáván pod open-source licencí.

Prezentační aplikace tvoří poslední část pomyslného toku dat od meteorologické stanice až k uživateli a je završením této práce. Vzhled stránky prezentační aplikace

je uveden na obr. 3.6, kde jsou uvedeny grafy závislosti teploty a atmosferického tlaku na čase za období 18.1.2018 – 19.1.2018. Vykreslené grafy zobrazují komplexní pohled na vývoj meteorologických veličin v místě stanice a splňují záměr praktické části práce.



Obr. 3.6: Stránka s meteorologickými daty ve formě grafů.

## 4 ZÁVĚR

Práce se v první kapitole zaměřuje na stručný popis LPWAN technologií, které jsou nyní k dispozici a až na čistě technické vlastnosti se nesnaží jednotlivé technologie mezi sebou porovnávat. Čtenář v první kapitole získá obecný přehled o nízkoenergetických a nízkorychlostních bezdrátových sítích. Tyto informace nemají vést k srovnání jednotlivých LPWAN technologií, ale k objasnění požadavků, které na bezdrátové síti klade obrovské množství zařízení, jehož nasazení se plánuje v rámci několika příštích let.

Druhá kapitola čtenáře seznamuje s problematikou více do hloubky a popisuje metody, kterými k problematice přistoupila konkrétní technologie LoRaWAN. Mezi tyto metody například patří modulace s rozprostřeným spektrem, vzájemná ortogonalita činitelů rozprostření a další. Z druhé kapitoly však již začínají vyplývat některé limitace, kterými se LoRaWAN technologie vyznačuje a to i přesto, že byla pro IoT specificky navržena.

Stěžejní částí práce je popis realizace zařízení využívající LoRaWAN síť. Tato kapitola popisuje využití konkrétních technologií dostupných v průmyslu k vytvoření meteorologické stanice. Záměrem bylo vytvořit systém, který měří meteorologické veličiny jako teplotu, tlak a vlhkost, je možné jeho stanice provozovat mimo prostory budov, bez běžných zdrojů elektrické energie a který je schopný změřená data vhodně prezentovat uživatelům systému.

Práce zároveň slouží jako studie proveditelnosti ověřující možnost vytvoření systému, který splňuje popsané požadavky. Za tímto účelem byla v rámci práce připravena meteorologická stanice, která je určena k odečtu meteorologických dat a aplikační server, který změřená data zpracovává, uchovává a prezentuje ve vhodné grafické podobě uživatelům. Poslední kapitola práce se zabývá konkrétními problémy spojenými s návrhem meteorologického systému, předkládá čtenáři možná řešení dílčích komponent systému a ke zvoleným řešením přidává odborný komentář.

Výstupem práce je plně funkční meteorologický systém, který potvrzuje výhodnost využití LoRaWAN radiové technologie v meteorologické aplikaci, která klade důraz na nízkou pořizovací cenu systému, možnost využití velkého množství měřících zařízení a dlouhou životnost bateriově napájených stanic. Navržený meteorologický systém by po rozšíření některých jeho částí byl vhodným kandidátem pro komerční využití, jelikož se oproti stávajícím meteorologickým systémům vyznačuje výrazně vyšší geografickou hustotou měřících stanic.

# LITERATURA

- [1] EVANS, D. *The Internet of Things: How the Next Evolution of the Internet is Changing Everything*. Cisco Internet Business Solutions Group (IBSG) [online]. 2011, [cit. 2017-10-25]. Dostupné z: <[https://www.cisco.com/c/dam/en\\_us/about/ac79/docs/innov/IoT\\_IBSG\\_0411FINAL.pdf](https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf)>.
- [2] ANDERSON, J.; RAINIE, L. *The Internet of Things Will Thrive by 2025*. Pew Research Center's Internet & American Life Project [online]. 2014, [cit. 2017-10-25]. Dostupné z: <[http://assets.pewresearch.org/wp-content/uploads/sites/14/2014/05/PIP\\_Internet-of-things\\_0514142.pdf](http://assets.pewresearch.org/wp-content/uploads/sites/14/2014/05/PIP_Internet-of-things_0514142.pdf)>.
- [3] *ESP8266 Power Consumption*. ESP8266 Developer Zone [online]. 2015, [cit. 2017-10-26]. Dostupné z: <<http://bbs.espressif.com/viewtopic.php?t=133>>.
- [4] BELLALTA, B.; *IEEE 802.11ax: Wireless Networking in High-density WLANs*. IEEE Spectrum Webinar [online]. 2017, [cit. 2018-5-24]. Dostupné z: <<http://www.dtic.upf.edu/~bbellalta/Bellalta-80211ax-IEEESpectrumWebinar8March2017-40mins.pdf>>.
- [5] BELLALTA, B.; BONONI L.; BRUNO R.; KASSLER A. *Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges*. Computer Communications, Elsevier [online]. 2016, [cit. 2018-5-24]. Dostupné z: <<http://www.iit.cnr.it/sites/default/files/1-s2.0-S0140366415003874-main.pdf>>.
- [6] *IEEE 802.15 WPAN™ Task Group 4 (TG4)*. IEEE 802.15.4 [online]. 2003, [cit. 2017-10-26]. Dostupné z: <<http://www.ieee802.org/15/pub/TG4.html>>.
- [7] *EMB-ZRF212B – 868 / 915 MHz ZigBee / 802.15.4 wireless module*. Embit - embedded and wireless solutions [online]. 2013, [cit. 2017-12-2]. Dostupné z: <<http://www.embit.eu/products/wireless-modules/emb-zrf212b/>>.
- [8] *A Cellular-type Protocol Innovation for the Internet of Things*. M<sup>2</sup>Communication [online]. 2015, [cit. 2017-10-26]. Dostupné z: <<https://www.theinternetofthings.eu/sites/default/files/%5Buser-name%5D/M2C%20Whitepaper%20for%20IoT%20Connectivity.pdf>>.
- [9] *A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications*. Texas Instruments Incorporated [online]. 2017, [cit. 2018-5-24]. Dostupné z: <<http://www.ti.com/lit/ds/symlink/cc2530.pdf>>.

- [10] *MR24J40 Data Sheet - IEEE 802.15.4 2.4 GHz RF Transceiver*. Microchip Technology Inc [online]. 2010, [cit. 2018-5-24]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>>.
- [11] *Data Sheet: JN516x - IEEE802.15.4 Wireless Microcontroller*. NXP Laboratories UK Ltd [online]. 2013, [cit. 2018-5-24]. Dostupné z: <<https://www.nxp.com/docs/en/data-sheet/JN516X.pdf>>.
- [12] *Radio Technology Overview*. Sigfox [online]. 2017, [cit. 2017-10-26]. Dostupné z: <<https://www.sigfox.com/en/sigfox-iot-radio-technology>>.
- [13] *Verizon carries successful data session on new NB-IoT Guard band network* [online]. Ericsson Research, Ericsson AB, 2018 [cit. 2018-5-24]. Dostupné z: <<https://www.ericsson.com/en/news/2018/1/verizon-nb-iot-guard-band-network>>
- [14] ERIC WANG Y.-P. et al. *A Primer on 3GPP Narrowband Internet of Things (NB-IoT)* [online]. Ericsson AB, 2017 [cit. 2017-10-30]. Dostupné z: <<https://arxiv.org/ftp/arxiv/papers/1606/1606.04171.pdf>>
- [15] *Vodafone jako první v ČR aktivoval NB-IoT síť*. LUPA.CZ [online]. 2017, [cit. 2018-5-24]. Dostupné z: <<http://www.dsl.cz/clanky/vodafone-jako-prvni-v-cr-aktivoval-nb-iot-sit>>.
- [16] KRUPKA, L.; VOJTECH, L.; NERUDA, M. *The Issue of LPWAN Technology Coexistence in IoT Environment* [online]. Department of Telecommunication Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 16627 Prague, Czech Republic, 2016 [cit. 2017-10-30]. Dostupné z: <<http://ieeexplore.ieee.org/document/7827866/>>
- [17] ADELANTADO, F.; VILAJOSANA, X.; PEIRO, P.; MARTINEZ, B.; SEGUÍ, J.; WATTEYNE, T. *Understanding the Limits of LoRa WAN* [online]. 2017 [cit. 2017-10-30]. Dostupné z: <<https://arxiv.org/pdf/1607.08011.pdf>>
- [18] LAMPE, J.; IANELLI, Z. *Introduction to Chirp Spread Spectrum (CSS) Technology* [online]. Nanotron Technologies, 2003, [cit. 2017-10-30]. Dostupné z: <[http://www.ieee802.org/802\\_tutorials/03-November/15-03-0460-00-0040-IEEE-802-CSS-Tutorial-part1.ppt](http://www.ieee802.org/802_tutorials/03-November/15-03-0460-00-0040-IEEE-802-CSS-Tutorial-part1.ppt)>
- [19] POOLE, I. *LoRa RF Interface* [online]. 2016, [cit. 2017-11-26]. Dostupné z: <[https://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/2\\_Humidity\\_Sensors/Sensirion\\_Humidity\\_Sensors\\_SHT3x\\_Datasheet\\_digital.pdf](https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/2_Humidity_Sensors/Sensirion_Humidity_Sensors_SHT3x_Datasheet_digital.pdf)>.

- [20] *SHT3x Datasheet digital*. Sensirion AG [online]. 2017, [cit. 2017-11-26]. Dostupné z: <[https://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/2\\_Humidity\\_Sensors/Sensirion\\_Humidity\\_Sensors\\_SHT3x\\_Datasheet\\_digital.pdf](https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/2_Humidity_Sensors/Sensirion_Humidity_Sensors_SHT3x_Datasheet_digital.pdf)>.
- [21] *BMP180 Specification*. Robert Bosch GmbH [online]. 2017, [cit. 2017-11-25]. Dostupné z: <[https://www.bosch-sensortec.com/bst/products/all\\_products/bmp180](https://www.bosch-sensortec.com/bst/products/all_products/bmp180)>.
- [22] TELKAMP, T.; KOOLIJMAN, M. *arduino-lmic - GitHub Repository*. [online] 2015, [cit. 2017-12-10]. Dostupné z: <<https://github.com/matthijskooijman/arduino-lmic>>.
- [23] *Wire Library*. [online] Arduino S.r.l. 2018, [cit. 2018-25-5]. Dostupné z: <<https://www.arduino.cc/en/Reference/Wire>>.
- [24] *arduino-sht - GitHub Repository*. [online] Sensirion AG 2016, [cit. 2018-25-5]. Dostupné z: <<https://github.com/Sensirion/arduino-sht>>.
- [25] *Adafruit-BMP085-Library - GitHub Repository*. [online] Adafruit 2011, [cit. 2018-25-5]. Dostupné z: <<https://github.com/adafruit/Adafruit-BMP085-Library>>.
- [26] GAMMON, N. *Power saving techniques for microprocessors*. [online] 2015, [cit. 2018-27-5]. Dostupné z: <<http://www.gammon.com.au/power>>.
- [27] *Low-Power - Low Power Library for Arduino*. [online] Rocket Scream Electronics 2016, [cit. 2018-27-5]. Dostupné z: <<https://github.com/rocketscream/Low-Power>>.



## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

SoC	systém na čipu - System on Chip
IoT	internet věcí - Internet of Things
LTE	Long Term Evolution
LPWAN	Low-Power Wide-Area Network
SoM	systém na modulu - System on Module
NB-IoT	NarrowBand IoT
ISO/OSI	mezinárodní organizace pro normalizaci / propojení otevřených systémů - International Organization for Standardization / Open Systems Interconnection
OFDMA	Orthogonal Frequency Division Multiple Access
3GPP	3rd Generation Partnership Project
IEEE	Institut pro elektrotechnické a elektronické inženýrství - Institute of Electrical and Electronics Engineers
LoRa	Long Range Radio
LoRaWAN	Long Range Radio Wide Area Network
SF	činitel rozptřeni - Spreading Factor
CSS	Chirp Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
DSSS	Direct Sequence Spread Spectrum
CRC	cyklický redundantní součet - Cyclic Redundancy Check
MCU	jednočipový počítač - Microcontroller Unit
OSHW	otevřený hardware - Open-source Hardware
OSS	otevřený software - Open-source Software
USB	Universal Serial Bus
SPI	Serial Peripheral Interface
I <sup>2</sup> C	Inter-Integrated Circuit
SCLK	hodinový signál - Serial Clock
MOSI	master výstup, slave vstup - Master Output Slave Input
MISO	master vstup, slave výstup - Master Output Slave Input
SS	Slave Select
SCL	hodinový signál - Serial Clock Line
SDA	datový signál - Serial Data Line
ABP	Activation-by-Personalisation
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
TCP	Transmission Control Protocol
SNR	odstup signálu od šumu - Signal to Noise Ratio

SRAM	statická paměť - Static Random-Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
DRAM	dynamická paměť - Dynamic Random-Access Memory
GPIO	všeobecně použitelný vstup/výstup - General Purpose Input/Output
UART	asynchronní sériové rozhraní - Universal Asynchronous Receiver-Transmitter
SMD	součástka pro povrchovou montáž - Surface-Mount Device
SQL	strukturovaný dotazovací jazyk - Structured Query Language
API	aplikační rozhraní - Application Programming Interface
URL	jednotná adresa zdroje - Uniform Resource Locator
ORM	objektově relační mapování - Object-relational mapping

# SEZNAM PŘÍLOH

A	Meteorologická jednotka	44
B	Aplikační server	45
C	Interpretace a zobrazení dat	47

# A METEOROLOGICKÁ JEDNOTKA

Výpis A.1: Nastavení v kódu meteorologické jednotky.

```
1 #include <lmic.h>
2 #include <hal/hal.h>
3 #include <SPI.h>
4 #include <Wire.h>
5 #include <Adafruit_BMP085.h>
6
7 Adafruit_BMP085 bmp;
8
9 // Nastavení klíčů a unikátní adresy koncového zařízení
10 static const PROGMEM u1_t NWKKEY[16] = { 0x2B, 0x7E, 0x15, 0x16, 0
    ↪ x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x19, 0xFF, 0
    ↪ x4F, 0x3C };
11 static const u1_t PROGMEM APPKEY[16] = { 0x2B, 0x7E, 0x15, 0x16, 0
    ↪ x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x19, 0xFF, 0
    ↪ x4F, 0x3C };
12 static const u4_t DEVADDR = 0x03FF0331;
13
14 // Používá se jen pro OTAA
15 void os_getArtEui (u1_t* buf) { }
16 void os_getDevEui (u1_t* buf) { }
17 void os_getDevKey (u1_t* buf) { }
18
19 static osjob_t sendjob;
20
21 // Jak často se má zpráva posílat?
22 const unsigned TX_INTERVAL = 240;
23
24 // Přiřazení funkcí pinům
25 const lmic_pinmap lmic_pins = {
26     .nss = 6,
27     .rxtx = LMIC_UNUSED_PIN,
28     .rst = 5,
29     .dio = {2, 3, 4},
30 };
```

## B APLIKAČNÍ SERVER

Výpis B.1: Směrovač HTTP dotazů knihovny Flask.

```
1 import json
2
3 from flask import request, jsonify, abort
4 from datetime import datetime
5
6 import app.database.models as models
7 from app.database.connector import db_session
8 from app.nodeapi import flask_app
9
10 @flask_app.teardown_appcontext
11 def shutdown_session(exception=None):
12     """ Remove scoped session after request """
13     db_session.remove()
14
15 @flask_app.route("/test")
16 def test():
17     return "OK"
18
19 @flask_app.route("/nodeapi/start/<string:date_start>/end/<string:
    ↳ date_end>", methods=["GET"])
20 def sensor_alarm(date_start, date_end):
21     try:
22         date_start = datetime.strptime(date_start, '%Y-%m-%d')
23         date_end = datetime.strptime(date_end, '%Y-%m-%d')
24         date_end = date_end.replace(day=date_end.day+1)
25     except:
26         abort(400)
27
28     zaznamy = db_session.query(models.Zaznam)\
29         .filter(models.Zaznam.timestamp >= date_start)\
30         .filter(models.Zaznam.timestamp < date_end).all()
31     response = [ [zaznam.timestamp.isoformat(), json.loads(zaznam.
    ↳ data)] for zaznam in zaznamy ]
32     #response.reverse()
33     return jsonify(response)
```

Výpis B.2: Databázové modely knihovny SQLAlchemy.

```
1 from sqlalchemy import Column, String, ForeignKey, DateTime, Integer,  
    ↳ DATE, cast  
2 from sqlalchemy.orm import relationship  
3 from sqlalchemy.ext.hybrid import hybrid_property  
4  
5 from app.database.connector import Base  
6  
7  
8 class Stanice(Base):  
9     __tablename__ = 'stanice'  
10  
11     id = Column(  
12         'id', Integer, primary_key=True)  
13  
14     zaznamy = relationship(  
15         "Zaznam", back_populates="stanice")  
16  
17 class Zaznam(Base):  
18     __tablename__ = 'zaznam'  
19  
20     id = Column(  
21         'id', Integer, primary_key=True)  
22     id_stanice = Column(  
23         'id_stanice', Integer, ForeignKey('stanice.id'))  
24     timestamp = Column(  
25         'timestamp', DateTime)  
26     data = Column(  
27         'data', String(250))  
28  
29  
30     stanice = relationship(  
31         "Stanice", back_populates="zaznamy")  
32  
33     @hybrid_property  
34     def date_timestamp(self):  
35         return cast(self.timestamp, DATE)
```

## C INTERPRETACE A ZOBRAZENÍ DAT

Výpis C.1: HTML s JavaScript kódem pro zobrazení zpráv z meteorologické jednotky.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Meteorologická stanice</title>
5 </head>
6 <body>
7   <p id="status">Připojuji se k serveru...</p>
8   <br>
9   <p id="init">Čekám na data...</p>
10  <p id="t">Teplota: C</p>
11  <p id="p">Atmosferický tlak: Pa</p>
12  <p id="snr">SNR: dB</p>
13  <script>
14    var token = 'vnoPhQAAAA1ldTEubG9yaW90LmlvTIMZL5ZIHfkInCf6HdmRJQ==
15      ↪ ';
16
17    var url = 'wss://eu1.loriot.io/app?token={token}';
18
19    function hex2a(hexx) {
20      var hex = hexx.toString();//force conversion
21      var str = '';
22      for (var i = 0; i < hex.length; i += 2)
23        str += String.fromCharCode(parseInt(hex.substr(i, 2), 16))
24        ↪ ;
25      return str;
26    }
27
28    var ws = new WebSocket(url.replace('{token}', token));
29    ws.onopen = function() {
30      document.querySelector('#status').textContent = 'Připojen k
31        ↪ serveru';
32
33    };
34    ws.onclose = function() {
35      document.querySelector('#status').textContent = 'Odpojen ze
36        ↪ serveru';
37    };
38  </script>
39 </body>
40 </html>
```

```

32 ws.onmessage = function(e) {
33     console.log('onmessage', e);
34     var data = JSON.parse(e.data);
35     console.log('data', data);
36     if (data.cmd !== 'rx') return;
37
38     document.querySelector('#init').style.display = 'none';
39     var tmp = hex2a(data.data).split(",");
40     var t = tmp[0].split(":")[1];
41     var p = tmp[1].split(":")[1];
42     document.querySelector('#t').textContent = 'Teplota: ' + t + ' C'
43     ↪ ;
44     document.querySelector('#p').textContent = 'Atmosferický tlak: '
45     ↪ + p + ' Pa';
46     document.querySelector('#snr').textContent = 'SNR: ' + data.snr +
47     ↪ ' dB';
48     };
49     </script>
50 </body>
51 </html>

```